**Content**

1. Introduction to Git
2. Git installation on windows
3. Git Terminology & command
4. Github account creation
5. Git practice

**1.Git**

**What is version control system(vcs)?**

🡪Version control is a system that records changes to a file or set of files over time**.**

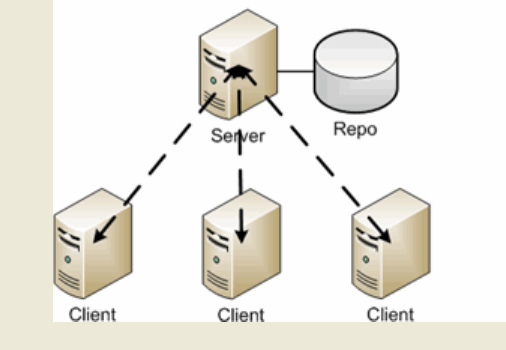
**What is Git?**

🡪Git was created by [**Linus Torvalds**](https://en.wikipedia.org/wiki/Linus_Torvalds) in 2005 for development of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel).

🡪Git is a [free and open source](https://git-scm.com/about/free-and-open-source) distributed version control system for tracking changes in computer files and coordinating work on those files among multiple people.

**Difference between cvcs and dvcs?**

**Centralized Version Control system(cvcs):**

****

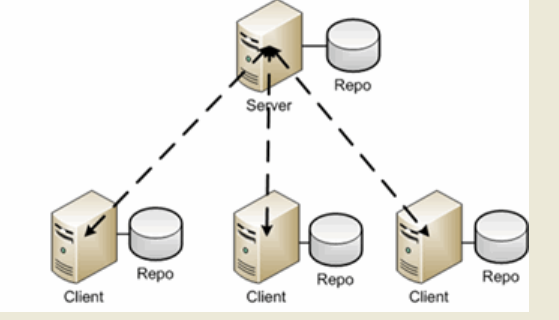
**Fig:** Centralized Version Control system

🡪 There is a single, central copy of the project content and history to which all  
users must refer.

🡪Typically accessed over a network, if the central copy is unavailable for some reason, all users are stuck.

🡪They cannot use version control until the central copy is working again

**Distributed Version Control system(DCVS):**

****

**Remote Repository**

**Local Repository**

**Fig:** Distributed Version Control system (DCVS)

🡪Distributed systems such as Git, on the other hand, have no inherent central copy.

🡪 Each user has a complete, independent copy of the entire project history,  
called a “repository,” and full access to all version control facilities.

🡪Network access is only needed occasionally, to share sets of changes among people working on the same project.

**2.Install Git On Windows:**

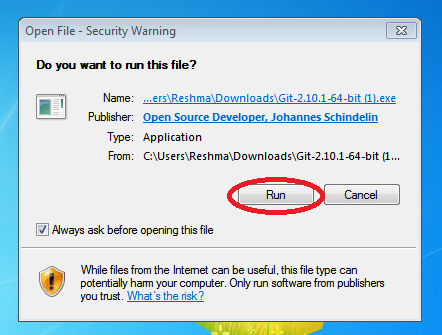
**Step 1**:

To download the latest version of Git, click on the link below:

***https://git-scm.com/downloads***

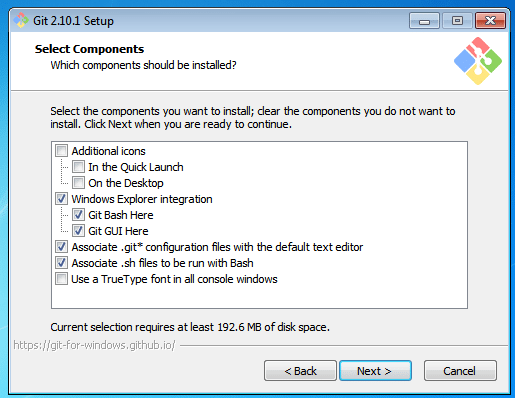
**Step 2:**

After your download is complete, **Run** the .exe file in your system.



**Step 3:**

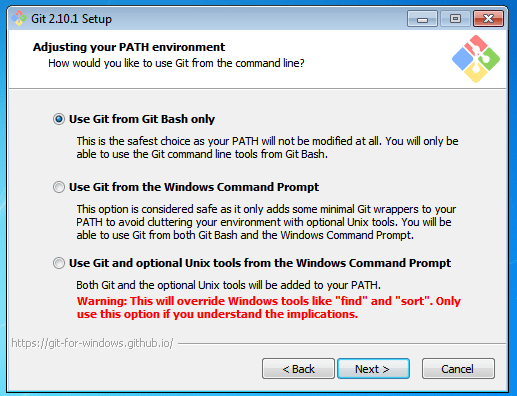
After you have pressed the **Run** button and agreed to the license, you will find a window prompt to select components to be installed.



After you have made selection of your desired components, click on **Next>**.

**Step 4:**

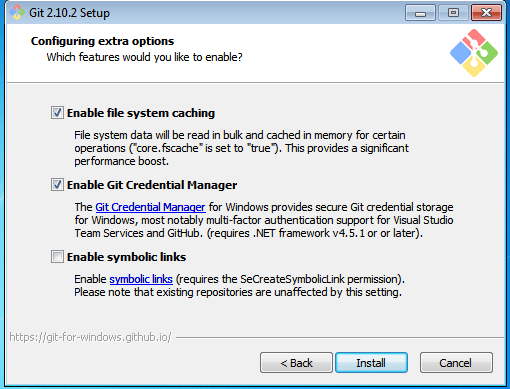
The next prompt window will let you choose the adjustment of your path environment. This is where you decide how do you want to use Git.



You can select any of the three options according to your needs. But for beginners, I recommend using **Use Git From Git Bash Only**

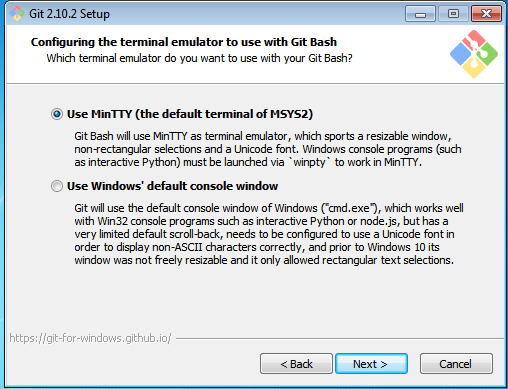
**Step 5:**

The next step is to choose features for your Git. You get three options and you can choose any of them, all of them or none of them as per your needs. Let me tell you what these features are:



**Step 6:**

Choose your terminal.



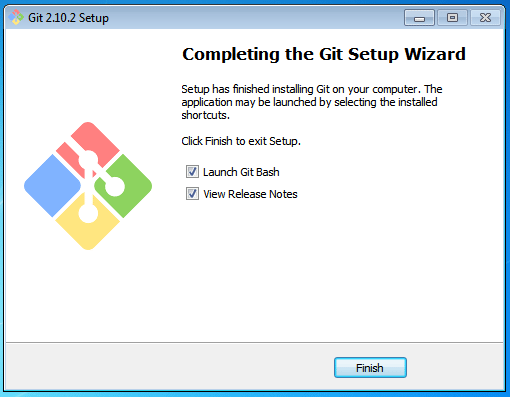
You can choose one from the options.

The default terminal of MYSYS2 which is a collection of GNU utilities like bash, make, gawk and grep to allow building of applications and programs which depend on traditionally UNIX tools to be present.

Or you can choose the window’s default console window (cmd.exe).

**Step 7:**

Select **Launch Git Bash** and click on **Finish**.



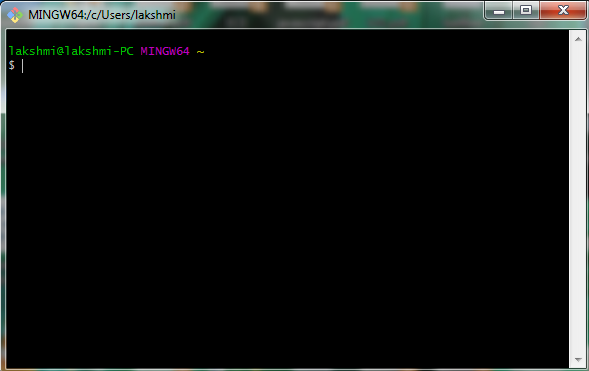
This will launch Git Bash on your screen which looks like the snapshot below:



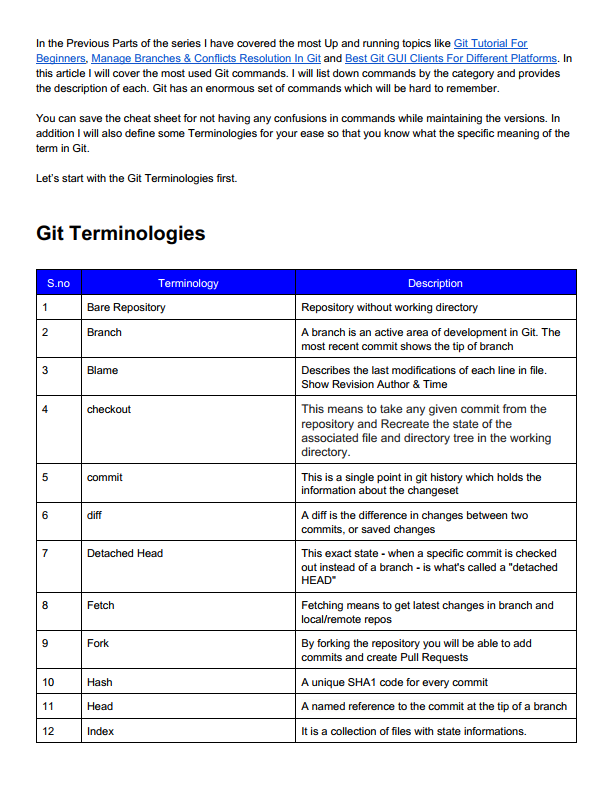
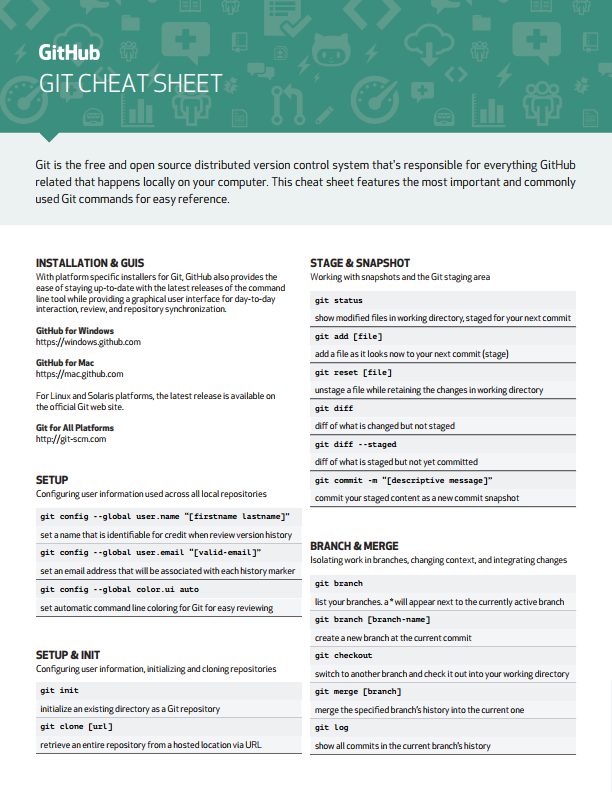
🡪double click on the icon it will open the below window

Or

Windows+r-🡪cmd🡪git --version



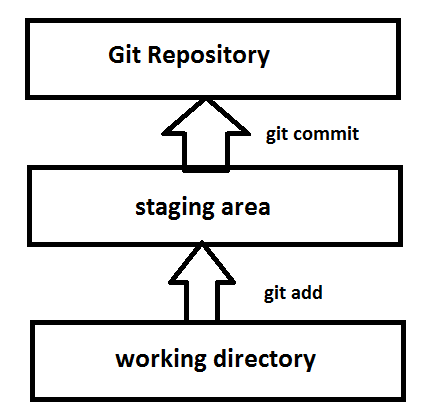
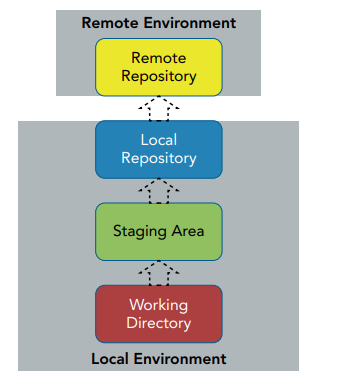
**3.Git Terminology:**

Git cheat sheet

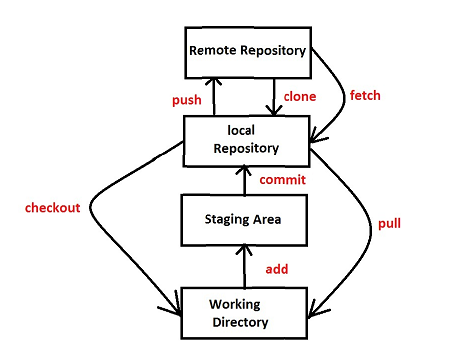
**Levels of Git:**

•Committed means that the data is safely stored in your local database.  
• Modified means that you have changed the file but have not committed it to your database yet.  
• Staged means that you have marked a modified file in its current version to go into your next commit snapshot.



1. You modify files in your working tree.  
2. You selectively stage just those changes you want to be part of your next commit, which adds *only* those changes to the staging area.  
3. You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.

**The Core Git Commands for Moving Content:**



**Fig:** Git in single pic

**Local Repository:**

add: content from the working directory to the staging area.

commit: staging area to the local repository

checkout: retrieve content (as ﬂat fles) from the local repository into the  
working directory.

**Local Repository to Remote Repository:**

push : To synchronize changes from a local repository to the corresponding remote repository.

**Remote Repository to Local Environment:**

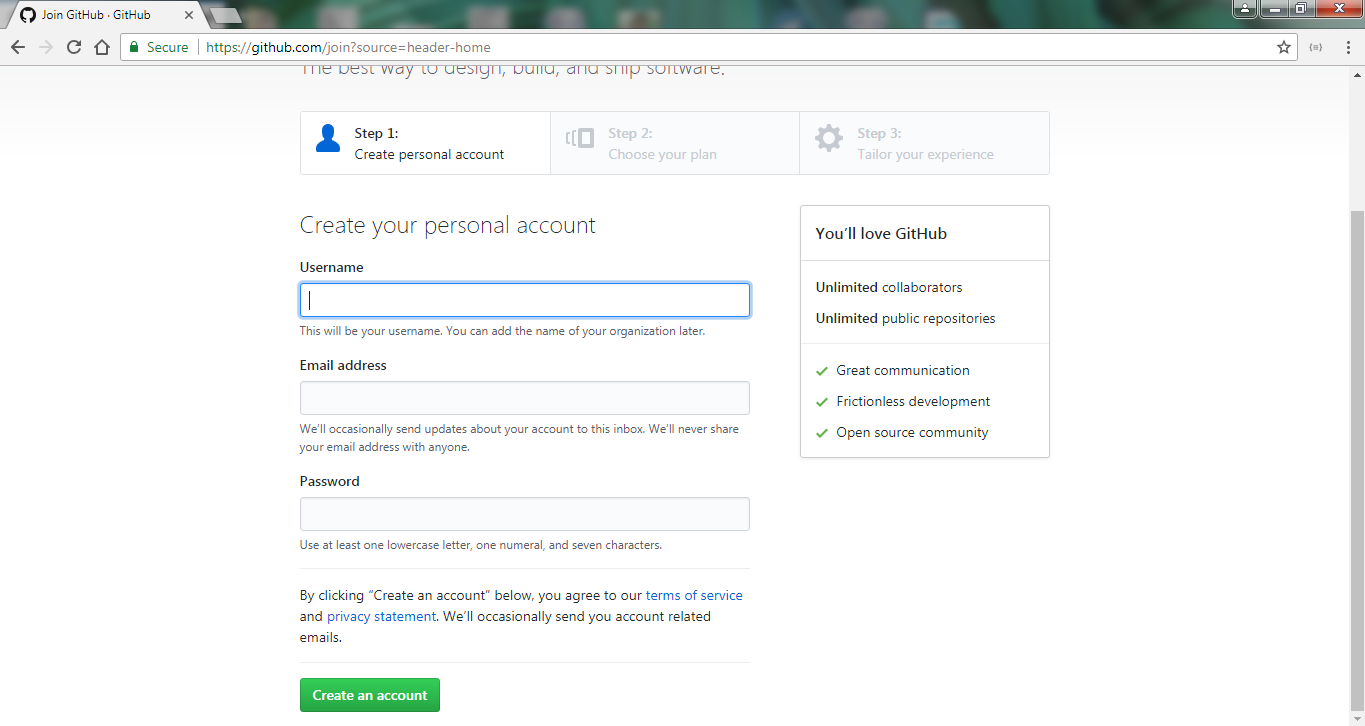
clone: it makes a local copy of the specified remote repository onto the local.

fetch: update the local repository from the remote repository

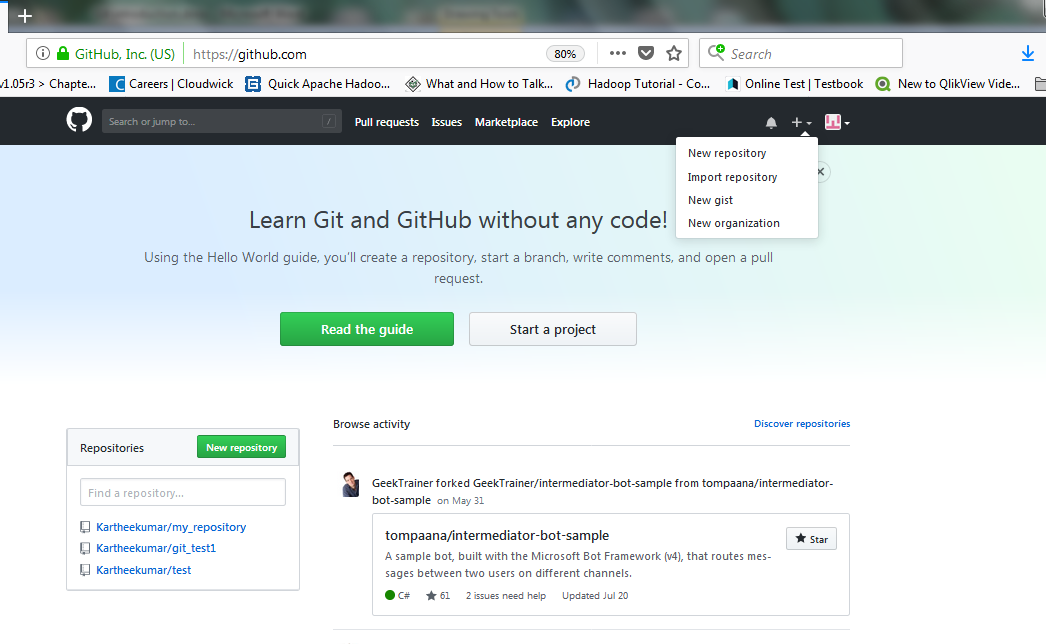
pull =fetch + merge: Fetches and merges to local branch and working directory

**4.Github account creation:**

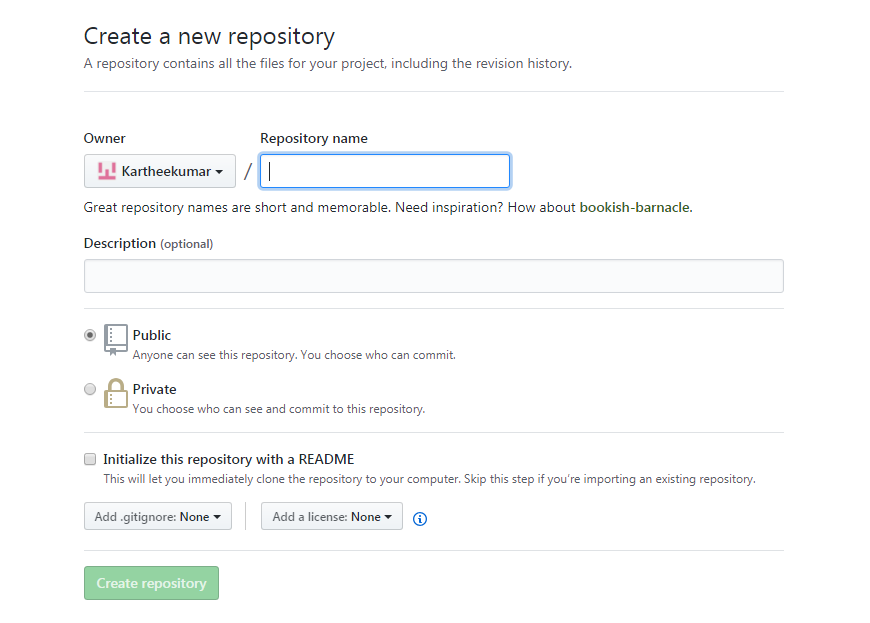
url: https://github.com/join?source=header-home



Creating new repository in Github



**or**



**5.Git examples:**